DOCUMENT RESUME

ED 305 641 CS 211 752

AUTHOR LeBlanc, Paul
TITLE Competing Ideologies in Software Design for Computer-Aided Writing Instruction.
PUB DATE Mar 89
NOTE 14p.; Paper presented at the Annual Meeting of the Conference on College Composition and Communication (40th, Seattle, WA, March 16-18, 1989).
PUB TYPE Viewpoints (120) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Software Reviews; *English Departments; Higher Education; *Politics of Education; Word Processing; Writing Instruction; *Writing Processes
IDENTIFIERS *Composition Theory; *Style Checkers (Software)

ABSTRACT

Composition researchers are poised to offer new answers to the three questions which are at the heart of ideological definition: what exists; what is good; and what is possible? Researchers must consider the impact of computer-assisted composition programs (often called "style checkers") as well as the basic word processor. Style checkers currently available place a strict emphasis on rules and mechanics of presentation and do not value content. The field must address the issues raised by entrenched product-centered software or it may find itself developing and working with tools uninformed by sound theory and pedagogy. Luckily, a grass-roots movement of teachers designing computer programs which emphasize quality of content has taken hold in the field of composition. The success of such ventures will depend upon the ability of software developers to gain recognition and support from English departments which typically see computers and those who work with them as part of the technological culture with which they will not conspire. Should English departments abdicate their responsibility to guide the development of computer tools for composition, the English teachers may very well find themselves working within an ideological, and thus pedagogical, concept of writing that is not of their own making. (Eighteen references are attached.) (RS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED305641

Competing Ideologies in Software Design for Computer Aided Writing Instruction

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Paul LeBlanc
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

By

Paul LeBlanc

Springfield College

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

My fourteen month old daughter subscribes to no particular ideology -- however, I was reading James Berlin's College English article on Rhetoric and Ideology at the same time she was using her fork not only for its intended purpose, but also to explore the VCR, bang the table, and prod the dog -- and I think she helped me understand the relationship of ideology and computers in new ways. If you'll forgive the homey analogy, Emma was using a tool to answer the three questions Berlin borrows from Goran Therborn and posits at the heart of ideological definition: What exists? What is good? and What is possible (479)? Of course, tools have always had that ideological power to force their users to reconceptualize their

CS211752

world and themselves. That's the message of Lewis Mumford's still wonderful 1934 book Technics and Civilization. Allow me to read one passage:

> When hunters acquired spears, for example, they must have seen themselves in an entirely new relationship to their world. Large animals which had earlier raided their foodstores and even attacked their children and which they feared, now became man's prey. Man's source of food grew, for now men could kill animals at a distance, including many species that had eluded them before. The effectively greater abundance of food must also have enlarged their domain over which they could range, thus increasing the likelihood that they would range, thus increasing the likelihood that they would meet other people. Their experience of the world changed and so too must have their idea of their place in it. (19)

The computer is the spear of our age. It is offering new answers to Therborn's three questions, and philosophers like David Bolter and Joseph Weizenbaum are attempting to describe the impact of the computer on our culture generally.

In composition, the use of the computer as a writing tool is very young -- while it might be said to have started with the commercial availability of the microcomputer in 1977, it was the introduction of the IBM PC in 1981 that gave it impetus. James Strickland's 1987 bibliography of research on composition and computers lists just 6 articles before 1982 and then 150 articles in the next five years. If the recent EDUCOM report on computing in postsecondary education is right in saying the mid-eighties will be

remembered as a time when there were "hardly any computers" on our college campuses (3), then we stand dramatically poised to offer new answers to "What exists?" "What is good?" and "What is possible?", perhaps reconstructing the ideology of composition.

As researchers trying to understand the impact of this new technology on writing behavior, we can no longer be solely concerned with the word processor as the computer's incarnation within the world of writing. The stand alone word processor is no longer a very interesting tool. I was one of the first instructors in the University of Massachusetts Computer-Based Writing Lab some five years ago, and one of the first things we all tried to do was to find ways to stretch the limitations of our word processors, by doing things like invisible writing and monitor switching (ala Stephen Marcus), electronic classroom journals (ala Dinan and Meeker), and electronic instructor feedback, through the laborious process of collecting and copying files onto student diskettes -- one or two at a time. We were like my daughter using her fork to take apart the VCR. We did not, by conscious decision, elect to use any of then available CAC programs, since they were mostly of the "drill and practice" variety or style checkers that reinforced a product-centered approach to composition pedagogy.

As you know, much has changed since then. When James Strickland wrote FREE in 1982, he did so because nothing like it existed at the time. When I requested a copy of it last year, he expressed embarassment of what he called its "primitveness." Indeed, written in BASIC with a modest requirement of 48K of memory, it is a distant ancestor from a program like Thoughtline, for

example, which requires a hard drive, 720k of memory and which, in its first version, was written in LISP. The lack of CAC software we felt in 1984 has been shortlived, and where we once could focus solely on the impact of word processing on the writing process, we are increasingly faced with a powerful array of new tools that have the potential to shape writing behavior each in their own way. As Ellen MacDaniel has pointed out, "As a profession, we would be mistaken not to keep this evolution of writing software under close surveillance; ultimately, its influence on texts, language, and thought may be as revolutionary as those of the writing and printing technologies that have preceded it. (140)"

To illustrate the implicit, if not necessarily intentional, ideological foundations of CAC software, I would like to compare two computer writing aids on the basis of the way they answer our ideological questions. I begin with Grammatik II, the latest version of the first style checker. I do so first because the distributor of the program tries very hard to sell the program as something more than a stylechecker -- here is a passage from the introduction to the manual:

> It does so much, the term "style checker" scorns its power. That's why we resist calling Grammatik II a "style checker." Why not call it an "electronic editor." (Bauer 1)

and I include it because I had the opportunity to observe its use for many hours in the local offices of a state agency that had recently gone on-line and had purchased Grammatik II as their sole writing aid beyond word processing.

The program runs a document through the following eleven functions:

1. Proofreads for typos.
2. Finds punctuation and capitilization errors.
3. Finds "clumsy, hackneyed, illiterate, trite, misused, pretentious, redundant, and wordy phrases..."
4. Offers readability analysis (Flesch Grade Level).
5. Offers count and percentage of passive voice and prepositions.
6. Compares your text to "Gettysburg Address." a Hemingway short story, and an insurance form, on the basis of: readability (grade level); # of words per sentence; # of letters per word; % of sentences in passive voice; and prepositions as a % of all words.
7. List of all words in your document (by frequency or alphabetical).
8. Dictionary of phrases.
9. Creates specialized dictionaries.
10. Will read document for special problems such as: archaic or foriegn words; vague adverbs; gender specific words; and redundancies.
11. Will mark problems in your text if you wish it to do so.

The manual says, "these include the same basic rules your English teacher tried to teach you." We all recognize the product-based criteria for writing evaluation implicit in such a list. They are part of what Henry Giroux would call an Instrumental Ideology, one which places a strict emphasis on rules and mechanics of presentation and form and which does not value content (209). Giroux argues that the underlying principle of instrumental ideology is the notion that human knowledge can be quantified, can be reduced to mathematical terms. This same notion, echoing Leibniz's seventeenth-century belief in a calculus of reasoning, is the cornerstone of modern computer programming, manifested in the work of Babbage, Boole, Turing, and in Claude Shannon's successful breakthrough attempts to transform logical reasoning into electro-mechanical circuitry. Because the kind of knowledge

represented on the screen lends itself most easily to computational rendering, it is no surprise that so many of the early writing aid programs should be style checkers. This is, after all, just the kind of thing the computer can do reasonably well.

What becomes interesting and important for us in the field, is that the mere ability to perform the task well gives it a kind of viability that flies in the face of our better knowledge. How else do we explain the investment of resources and energy into sophisticated "second generation" programs like IBM's CRITIQUE, which generates grammar and style critiques, even as almost everyone in the field of composition agrees that "process" is good and "current-traditional" rhetoric is bad, to use Lester Faigley's succinct summary of the paradigm shift (527)? Or is this a case of the tool, which has the ability to perform certain tasks better than others, answering our ideological question of "what is good?" in a way different than we might answer it without the tool? Joseph Weizenbaum has argued that computers often serve to "entrench and stabilize social and political structures" that might otherwise be reformed or radically altered (31). Programs like Grammatik II and Critique can be seen as serving such a function, reinforcing a still widespread understanding of literacy that Giroux ties to state needs, as illustrated in his examination of UNESCO literacy programs in developing nations (215). Andrew Sledd recognizes this ideological link when he associates CAI programs with back to basics reactionaryism in his 1988 College English article on literacy (499). These are the kinds of issues we must address, or we may find ourselves developing and working with tools uninformed by sound

theory and pedagogy, and perhaps possessing an ideological foundation we would prefer to keep out of our classrooms.

Happily, the grass-roots movement of teachers designing computer programs, an educational movement that John Kemeny and the developers of BASIC hoped for in 1958, has taken hold in the field of composition. So many of the best CAI programs now available to us have come out of the composition classroom, and that continues to be the case. One such program is INTERCHANGE, a program that came out of Paul Taylor and the Deadulus Group's work at the University of Texas and was in part inspired by Trent Batson's work at Gaulladet University. INTERCHANGE is a real-time conversation program using a standard file-transfer microcomputer network to create on-line classroom discussion. It has five main components:

1. The individual student writes on a "scratch pad", and when ready, can send the comment into the main display window on every screen signed-on to the conference.

2. Comments appear on the main window as they arrive in chronological order and the screen scrolls to make room for new incoming comments. Students can move through the text to reread what has been sent.

3. Users can break off from the main conference and create sub-conferences to pursue new topics or lines of conversation.

4. It has a split screen capability that allows a conversation to take place on one screen about text that appears on another window.

5. The program can provide a hardcopy transcript of the conversation at the end of the session.

A program such as this provides a dramatically different set of answers to Therborn's three questions than does a program like Grammatik II. This program locates the real, the answer to the question "What exists?", in the dilaectic interaction of writers

working within a discourse community. It places positive value in egalitarian participation in social discourse, empowering participants who, for a variety of well documented reasons, might not otherwise participate fully or at all. It allows those participants to ignore many of the social/hierarchal cues that mark traditional classroom exchange -- cues associated with gender, race, position, social status, or appearance (Selfe 12-3). What matters in this program is the quality of ideas -- not the societal definition of those who produce them. I would argue that this is just the sort of literacy behavior Cindy Selfe envisions in her work on feminist theory and computer technology (7). While Grammatik II serves a current-traditional rhetoric, Interchange goes a long way towards realizing the ideal social-epistemic classroom Berlin envisions at the end of his article, a classroom which welcomes, in his words:

> behavior that is always open-ended, receptive to the unexpected, and subversive of the planned. Most important, success in this classroom can never be guarenteed. This is a place based on dialectical collaboration -- the interaction of student, teacher, and shared experience within a social, interdisciplinary framework -- and the outcome is always unpredictable. (492)

The program will not achieve this alone, yet it is a powerful tool which opens up new possibilties for the composition classroom.

However, not everyone is so optimistic about the future of CAC software. Nancy Kaplan, a co-designer of PROSE, winner of an EDUCOM Distinguished Software Award in 1987, asserted in a recent interview with me that:

> in a sense, what we are working with now, I think, was shaped for us in the 50s and 60s and we weren't there. What is being shaped now is what's going to appear on our desks in 10-15

> years. And we're still not there. We're still not where the shaping is going on. Just a few of us are. (p. 17, lines 22-7)

In many ways, we are struggling to work with tools that were, for lots of reasons, designed without our input. Kaplan recognizes the danger of that -- it is the phenomenon that Mumford illustrated about tool use or technology, which is that the tools we use force us to reconceptualize what we do, and once that happens, there's almost no going back. The question then, is how do we as a field play a more active role in CAC design -- in shaping the tools we and our students will work with in the future.

Our success in that venture will depend in large part upon our ability to first convince our colleagues in the field of composition that computer technology stands to dramatically alter our understanding of what we do. We then must go one step further and convince English Departments to support and reward those who work in CAC design. In my interviews with program designers, it became very clear that with the exception of large technically oriented research institutions like CMU, English departments have failed to adequately support and reward CAC researchers. If composition researchers are still struggling for recognition within departments, computer and composition researchers are more marginal yet. A few examples from my interviews. In one case, the designer of a good and useful program used nationally in the public schools recieved no professional recognition within her institution and continues to work as an adjunct faculty member on a year to year contract. In another case, a university challenged the ownership of programs designed within their computer facilties and effectively drove CAC

designers off the campus to form their own outside CAC research and design group. In another case, a designer's work was not only unsupported in terms of money, hardware, or release time, but it was also made clear to her that her CAC work was to take place on her own time and was not to interfere with so called academic duties. By the way, the success of her program earned her no departmental recognition outside of a blurb in the newsletter and she too continues to work as an adjunct on a one-year renewable contract.

These program designers are losers in an ideological struggle of another kind -- one in which the power of tenure, promotion, and departmental resources are held in the hands of English Department humanists who see computers and those who work with them as part of the technological culture with which they will not conspire. It is the struggle that Ted Nelson examines in his quirky and often enlightening book Literary Machines between what he calls the Noids, short for technoids, and the Fluffies, the humanists. Nelson calls for a reconciliation of the two, to produce what he calls "Systems Humanists." He says:

> In the broader view, the goals are the long ones of civilization -- education, understanding, the preservation of human values -- but we must use today's technologies. I call this view systems humanism. Civilization as we know it is based in part on running water. That system had to be thought out. Similarly, somebody's gotta design waterworks for the mind. But it should be someone who understands the fluidity of thought. (1/14)

In his 1983 article "Understanding and Evaluating: The Humanist as Computer Specialist," Joseph H. Bourque makes a similar argument for the field of composition. He says:

> ...if we are to enjoy cooperation in achieving our common educational goals, we must recognize that it takes many kinds

> of efforts to make a successful English Department. Computers are an inevitable part of our future, and those who facilitate their use make a significant contribution to the discipline which must be acknowledged. (73)

Steven Jobs calls the computer the "finest tool" built by the finest tool builders who have ever lived (Sprecher 127). That tool will have great influence on our understanding of writing -- in ways we are only beginning to understand now. I have looked at only one small piece of that puzzle, the way CAC programs can promote an implicit rhetorical ideology, and I have found myself finally talking about English department politics -- but there is a vital connection between the two, between our role in the integration of computers in composition (with all kinds of implications for our understanding of textuality generally) and the willingness of English departments to answer the questions of "What exists?", "What is good", and "What is possible?" in new ways. Should English Departments abdicate their responsibility to guide the development of computer tools for composition -- tools we will surely be working with in the future -- we may very well find ourselves working within an ideological, and thus also pedagogical, concept of writing that is not of our own making.

WORKS CITED

Balestri, Diane Pelkus. Ivory Towers, Silicon Basements. McKinney, TX: Academic Computing Publications, 1988.

Bauer, Ron. Grammatik II, The Writing Analyst Instruction Manual. San Francisco: Reference Software. 1986.

Berlin, James. "Rhetoric and Ideology." College English 50 (5) (1988): 477-494.

Bolter, David J. Turing's Man: Western Culture in the Computer Age. Chapel Hill: UNC Press, 1984.

Bourque, J. "Understanding and Evaluating: The Humanist as a Computer Specialist." CE, 45 (1983).

Dinan, J.S., and R. Gagnon and J. Taylor. "Integrating Computers Into the Classroom: Some Guidelines." Computers and Composition 3 (2) (1986): 33-39.

Faigley, Lester. "Competing theories of Process: A Critique and Proposal." CE 48 (1986): 527-542.

Giroux, Henry A. Theory and Resistance in Education: A Pedagogy for the Opposition. South Hadley, MA: Bergin, 1983.

Kaplan, Nancy. Transcript of telephone interview on 2-1-89.

Marcus, Stephen. "Real Time Gadgets with Feedback: Special Effects in Computer-Assisted Instruction." The Writing Instructor 2 (1983): 174-181.

McDaniel, Ellen. "Bibliography of Text-Analysis and Writing Instruction Software." Journal of Advanced Composition 7(1 & 2) (1987): 139-169.

Meeker, M. "Waiting for WANDAH: A Critique of Present Trends in Computer Assisted Composition." Computer Assisted Composition Journal 1 (1) (1986): 42-54.

Mumford, Lewis. Technics and Civilization. New york: Harcourt Brace, 1934.

Nelson, Ted. Literary Machines. Theodor Nelson, 1987.

Selfe, Cynthia. "Technology in the English Classroom: Computers Through the Lens of Feminist Theory." In process. 1988.

Sprecher, Jerry W. Facilitating Academic Software Development. McKinney, TX: Academic Computing Publications, 1988.

Strickland, James. "Teaching Writing With Computers: An Annotated Bibliography." James Strickland, 1987.

Weizenbaum, Joseph. Computer Power and Human Reason: From Judgment to Calculation. New york: W. H. Freeman, 1976.